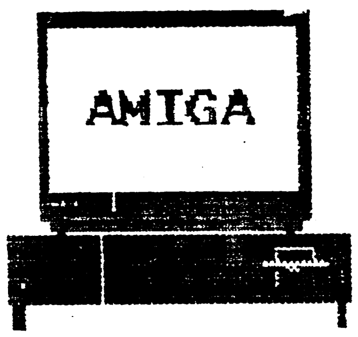
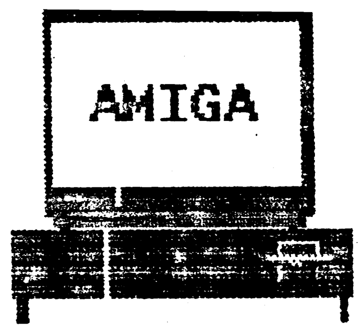


BECAUSE



BULLETIN OF THE CANBERRA AMIGA USERS SOCIETY

MEETINGS:

Meetings are held on the second Wednesday of each month on the second floor of the J.G. Crawford building at ANU, starting at 8:00 PM. Roll up any time after 7.30.

The next meeting will be held on the 10th of December.
The following meeting will be on the 14th of January.

SUBSCRIPTIONS:

Annual fees are $20, payable at any of the monthly meetings to the treasurer.

AIMS/BENEFITS:

CAUSe is an independent group formed to bring together people who own, use or are interested in the Commodore Amiga computer. Members receive a bi-monthly bulletin providing a wealth of information concerning the Amiga and are encouraged to attend our monthly meetings.

COMMITTEE:

Director: Wayne Myles --- 45 8761 (W) Secretary: Chris Townley
Treasurer: John Bishop --- 49 3786 (W) Editor: Craig Fisher - 54 6033 (H)
Auxiliary: Peter Whigham - 49 3786 (W) Auxiliary: Chris Foster - 80 6877 (W)
Auxiliary: Peter McNeil -- 54 2732 (H)

BULLETIN BOARD:

CAUSE has it's own FIDO-NET bulletin board which is available free of charge to anyone with a modem.
The board is Baud rate auto-sensing and will accept 300/300, 1200/1200 and 1200/75 baud rates.
Sysop: Mike Hurst-Meyers BBS: 59 1137 from 16/12/86, presently 58 4055

IN THIS ISSUE:

Editorial

Well here it is, the first issue of BeCAUSe. I'm very pleased with the material I've received for this newsletter - I think there should be something here to cater for everyone. Thanks very much to the people who have contributed to this issue who will, no doubt, do so again. To everyone else - your writings will be much appreciated by others. I would also like to hear from people telling me what they would like to see printed in the newsletter. Whether we keep the bulletin as a fairly large bimonthy or change it to a smaller monthly newsletter at some later stage is another matter for discussion.

One of the policies I've adopted in producing the bulletin is that everything will be printed on the Amiga (I declined an offer to use a Mac with PageMaker) - this way people will see how the availability of good Amiga text editing and printing software is improving and just what it can do. (At the moment we have only editors, pretty poor word processors and the printer drivers - there should be some desktop publishing software out RSN [Real Soon Now])

Enjoy this issue and start thinking about what you want to see in the next one which will be out in February.

Craig Fisher

Writing for BECAUSE

Writing for BECAUSE - what other reason do you need?

The next issue of BECAUSE will be in February so now is a good time for you to start writing articles for it. We want items on a whole range of topics to appeal to the many different Amiga users.

You can get your articles to me by giving either a disk containing the text file or a printable printout (i.e. fairly dark ribbon) to me or another committee member (who will give it to me). On the other hand, if you own or have access to a modem you can upload articles to the bulletin board, where there is a special file area for them, for me to download them.

Looking forward to a flood of articles...

Conventions used

In the pages of this publication, the following conventions will be used...
Ratings are assigned on these grounds:

Performance:	*****
Documentation:	*****
Ease of use:	*****
Value for money:	*****
Reality Coeff.:	0.0-1.0
Correctness: (lack of bugs)	*****

Ratings:

*	Terrible
**	Below Average
***	Acceptable
****	Good
*****	Excellent

Reality Coefficients:

0.1	Product has been advertised, but not seen
0.3-0.49	Product exists in "Beta Test" form
0.5-0.79	Has been SEEN to work by reliable witnesses
0.8-0.99	Available off-the-shelf
1.0	No bugs known, works as documented, off-the-shelf.

Doctor Byte at Large.

Hi Folks!

Since this is the first issue of BeCAUSe, I thought that the milling hordes that are now joining the Society might like to hear a bit about the history of the group.

The group has been running officially for just on three months now, but in a way, it has been in existence since last August, when we first heard about Commodore's New machine in the Byte preview they ran then.

The early interest in Amiga quite surprised me. There were quite a few "sight unseen" orders...including mine. In fact, very nearly all of the first shipment of machines were sold before they arrived in the country! I have been involved with other Micros and User Groups over the years, but these people were "only" keen...My experience of Amiga owners is that they (I) am (are) absolutely *rabid!* The missionary spirit rules, OK?

After what seemed to be an interminable wait, we the consumers finally got our offering from on high, the first Amigas. I remember being not a little depressed at hearing of C-A's Henry Fordism "You can have any monitor you want, as long as it's *this* one." I wanted a smaller monitor for games, and so on (Smaller monitors usually give better colour, due to the higher dot/line density), and a Monochrome monitor for "everything else". I must confess at this point to be somewhat of a Mac supporter...The polish on Mac software is (in my opinion) without equal at the moment, but I couldn't stand locking myself into an AppleCart forever more -- I like to chop and change manufacturers.

A week or so later, we finally got our external disk drives, and could start actually DOING something with our machines. If any of you souls out there are still suffering with a single drive, give *serious* thought to getting another. You will *never* look back. (I wanted three!)

About this time, I decided to start organising a User Group; The time was ripe; this was, and still is the time when they are needed the most, while the machine is just beginning to gain a following, and very little information on it was available.

In living up to the missionary springit, talking everyone around me into buying one of these wonderful boxes, I ran into a surprising number of people I knew *Who had already bought one!* Word got around, and we decided to hold the first meeting at my humble abode.

About seven turned up, several more couldn't make it (and still have difficulty making it, due to Wednesday night obligations), and had a great time exchanging ideas, suggestions, and information...not least about what parameters people had discovered on the AmigaDOS commands. They were at that time, and for two months afterward, an **Undocumented Feature** (usually read "Bug") of the system.

One of the important (to me) ideas to come out of the meeting was a reinforcement of my ideas about software ethics. Those present in the first meetings agreed, substantially at least, that piracy was a thing to be frowned upon in general, though it has a purpose in fact. Let me explain that remark.

I think it is my right to "crack" a protection scheme that gets in my way. I get very annoyed by such schemes. I also believe in **buying** software (as my shelf testifies), but I will not buy a program unless it is worth the price. Hence, I "acquire" (as the euphemism goes) rather a lot of software for "evaluation". When (if) it proves its worth, I buy it. Otherwise, it becomes disk drive fodder. I am constantly annoyed by people who "acquire" software and never actually pay for it. The Constitution of the Society embodies this principle of fairness to Software Authors.

I will not pay extortionate prices, either. This prompted me to start a very small importing operation, for specialised products that were unreasonably priced in Oz.

Quite a few copies of Lettuce C went out the door, and more copies of TDI Modula-2 than I would have thought possible.

By the third meeting, we had a semi-stable location for meetings, at the Computer Science Departmental Centre, at the ANU. By this time, we were all die-hard cynics. We came up with the Reality Coefficient, a curious concept that attempts to define the accessability of a product.

My suggestions at this point were: A product which has been advertised gets an RC of 0.1, One which exists in "Beta Test" form gets a "0.3", or "0.5" if it has actually been *seen* to work by someone trustworthy, 0.8 to 0.9 for a product on the shelf, and a product which exists on the shelf *and works as documented* gets the even 1.0.

As a general rule, I doubt if the first version of *anything* can get a "1", or indeed greater than "0.9" simply because Amiga programmers are so busy fighting the programming interface and marketing people to actually *finish* their creations! Naturally, these numbers can be interpolated given appropriate grounds (say, the all-important rumours, like the ones about the Amiga 2000 and 2500...).

By August, we had moved to the new home of ANU DCS, the JG Crawford Building, and we are hoping to be able to use this facility for the forseeable future. Thanks to Dr Robin Stanton (Head of DCS), Dr Brian Molinari (Lecturer, and Now Chair of CAUSe Meetings), and Peter Whigham (tutor) for the work they put into organising this.

We had our ears blown away by a demo of a Midi interface developed by members, and were intrigued by some el-cheapo disk drives developed by the same people (See elsewhere in this issue). All in good fun, and this was the germ of the Society as it is today. The turnout was well beyond expectation, about thirty, and we haven't looked back since. During the meeting, we organised a committee to write up a constitution for presentation to the next meeting.

September was a fairly dull meeting for all, but an important one none the less, for this was when our formal structure was created. After a somewhat heated debate, particularly from some parties who had plenty to say about the constitution *after* it had been drafted, not (of course) *during* the drafting. There was, however, good representation of those present, and most people got involved at one stage or another. This (eventually) being done, Office bearers were elected, and responsibilities assigned. A subcommittee was set up to investigate the chances of starting our own BBS (Bulletin Board System, a computerised message centre which you call up with a modem). Everyone left the meeting more than a little bedraggled.

October saw the introduction of the Annual Budget. This unfortunately occupied most of the meeting (again, due to dissenting elements). An interesting development (due to no little stirring up of interest) was that Steve Nimmo paid for, and Mike Hurst-Meyers and Alan Salmon set up a FidoNet BBS for us out of the goodness of their hearts! Thanks guys! This made a moot point of the BBS committee, so it was disbanded. There was some discussion of the new BBS.

These two meetings no doubt alienated a few people, but the fact remains that we now have one of the best organised User Groups around. Such Administrivia has a price though, and we have now paid in full. A budget allocation made way for the publication of the Rag you are now reading. Perhaps in a few months we shall be able to go monthly, but for the present, it will be published bimonthly.

The problem, in a word, is money. In order to keep the membership fee down to a reasonable level, we had to sacrifice frequency. When we get a good head of steam, and have some other revenue generating schemes underway, we will review this policy. It was decreed, and universally accepted, that the next meeting should be a *real* general meeting.

November saw the coming of age of the Society; the balance of the projected forty members of the Society paid their membership fees (somewhat more than were expected *this* early). The meeting was given over substantially to a demonstration of the FidoNet BBS.

If you have not yet been bitten by the BBS bug, I suggest you find yourself a modem and try it out. It's just like having a meeting that runs all month long! (except that *you* chose when you want to turn up). Come on, You've spent at least $2500 on a computer, a modem is less than a week's pay...get in on it! Imagine being able to send a message anywhere in the world for Nothing, free, gratis! (No, there is no catch.)

By this stage, the regular meeting size was 35-45. Meetings (according to the Constitution) are open to all, but the services of the group are only available to members. The BBS is open to anyone who is prepared to give their address and phone number to the Sysop (this information is confidential, and will *never* be used for any purpose other than contacting the person in question)

The Society is able to sell disks at very competitive prices. At latest count, we can get disks for $45 per box of ten. There is a profit of about 5% in this for the Society, to assist in expanding the services of the Society.

Other User Groups charge a fee of about ten dollars *per disk* for the duplication of Public Domain Software. At the moment, we have a good arrangement with Steve, who allows duplication of disks on his machines in the store. An idea which I have thrown around is the formalisation of this disk library, using it as another source of revenue for the Society. With 45-odd disks of public domain software, it *needs* some organisation.

I propose that the society buys a pile of discs, plus a carrying box, to put the PD software in; then this box can be passed around, each member making a donation to the Society concomitant with the amount of use he makes of it. In this way, we can not only circulate the new software sooner, but can afford to acquire it *faster* through our own channels (It costs about $Oz15 per disk to import them, keep that in mind when you copy your next batch).

BBS Users are no doubt aware of the UseNet listings which have been posted in the file area. These were laboriously collected by Dr Brian Molinari and downloaded to my PC at work to be sent to the Fido machine. These listings are a mixed bag; there is some dull stuff, but quite a lot of good stuff too. We are making arrangements to do the uploads in a continuous mode, instead of collecting a megabyte or so as we have been doing. Thanks, Brian! For those who don't know what Usenet is, it is somewhat akin to Fido, but an order of magnitude bigger, running on "Real" computers all over the world. Panic, Youse guys, we're catching up Real Fast!

Future meetings are essentially *carte blanche*. It would be appreciated if members could notify the committee at least a week in advance of suggested material for meetings (preferably at the previous meeting). I can be contacted during business hours on 45-8761, but prefer to keep my own time to myself whenever possible. I log on to the BBS as often as I possibly can, so if you can't get me on my work number, try there next (I am rarely far from the phone).

Anyway, I think that's enuf rambling from me. Let the others have a say, wot?

Oh, and I'm still waiting for a Database program and a decent WP program. Sigh.

TTFN, Doctor Byte.
(alias Wayne Myles)

NOTES ON A "C" PROGRAMMING STYLE

As an introduction to programming in "C" on the *Amiga* it is worthwhile to consider the overall file structures which should be used and some syntactic styles which will make the programs clearer to read and therefore easier to debug.

File Organisation Structure

As an example, consider a program with the following structure:

```
          ----------------------------------------------------------
        :                                                         :
main.o  : -----------> file1.o ------------------------------> C.lib
        :                                                         ^
        :                                                         :
        : -----------> file2.o -----------------------------------
        :                      <---
        :                         :
        :                         :
        : -----------> file3.o  ---
```

The arrows indicate a dependency in terms of functions/structures which are used between each file. ie. file3.o uses some feature declared in file2.o.

This structure translates to the following **#include** layout at the top of each "C" file:

main.c	file1.c	file2.c	file3.c
"stdio.h"	"stdio.h"	"stdio.h"	"stdio.h"
"ufile1.h"			"ufile2.h"
"ufile2.h"			
"ufile3.h"			

We have 3 classes of logical file groups:

1/ **Main** contains the startup and initialisation calls, and the main routine. This module becomes the **NAME** of the program. There is only one file at this level.

2/ **Abstract Data Type Level**

This level contains files that are to be linked to the program. Each module consists of two parts:

i) The User Header File

syntax : **u<filename>.h**

This file contains the data structures, externally available data variables, typedef declns., and functions available to the user of the file module.

ii)The Implementation File

syntax: <filename>.c

Contains the code (ie. functions) for the file. Note that much of the user header file is reproduced in this module, although no *extern* declerations are present.

EXAMPLE

```
FILE: file3.c

#include "stdio.h"
#include "ufile2.h"

< define statements >
< typedef declns.   >
< declerations      >
< code (functions)  >

FILE: ufile3.h

< define statements >
< typedef  declns.  >
< extern declns.    >
```

Further notes on Style: **#define & typedef statements**

All constants should be declared as

```
            #define <name> <value>
```

where <name> is UPPERCASE, and <value> corresponds to the value of <name>.

```
i.e.        #define MAXDAYS 100
```

All structures should be declared in the following example form:

```
typedef struct _date
{
      int day,
          mon,
          year;
}DATE,*DATEPTR;

typedef struct _days
{
      DATE futuredates[MAXDAYS]; /* array of DATE records */

}DAYS,*DAYSPTR;
```

Hence to declare a variable of type DAYS :

```
               DAYS thedays;
```

and to declare a pointer to a DAYS structure

```
               DAYSPTR daypointer;
```

Casting is performed in the following way

```
               bob = (DAYSPTR) getday();
```

which casts bob as a pointer to a DAYS structure.

These syntactic rules allow the code to be simple and clear since we can avoid the use of explicit typedef or struct references. The UPPERCASE format makes for easy reading and indicates clearly when structures are being referenced or declared.

In following issues this format will be used when giving examples of "C" code, and it would be hoped that user group members would attempt this *module* style of design so that libraries of useful routines can be set up for the groups usage. This section of the newsletter is also designed to answer questions on Programming in "C" on the *Amiga*. Send problems or comments to any committee member for publishing/response in forthcoming issues.

Wildcards in CLI -- Craig Fisher

(Information from the AmigaDos User's Manual)

It seems that a lot of people don't know that some CLI commands accept wildcards in filenames, or how to use them. Wildcards are special characters or sequences of characters which let you specify only part of a filename or filenames with a certain pattern. Some examples, given below, should make this clearer.

The CLI commands Copy, Delete, List and Search all allow you to use wildcards to match filenames. The special characters used for pattern matching are: '()?%#| and are used as follows.

Prefacing one of the special characters with ' (apostrophe) removes the effect of that character so you can use it as part of a filename.

?	matches a single character
%	matches the null string
#<p>	matches zero or more occurences of the pattern <p>
<p1><p2>	matches a sequence of pattern <p1> followed by pattern <p2>
<p1>|<p2>	matches if either pattern <p1> or pattern <p2> match
()	groups patterns together

A few examples of the use of wildcard pattern matching follow:

LIST PAT A#BC	Will list files AC, ABC, ABBC, ABBBC etc. (in the current directory only)
DELETE A#(B|C)D	Will delete files AD, ABD, ACD, ABCD, ABCBCD etc.
LIST PAT A?B	Will list files AAB, ACB, AFB, AZB, A2B etc.
SEARCH A#?B qw	Will search files AB, AXCB, A3FCNB, AMIGAB etc. for qw
LIST PAT '?#?'#	Will list files ?#, ?YTE#, ?A#, ?NMN# etc.
LIST PAT A(B|%)#C	Will list files A, ABC, AC, ACCC, ABCCC etc.
LIST P #(AB)	Will list files AB, ABAB, ABABAB, ABABABAB etc.
DELETE RAM:#? ALL	Will delete everything in the RAM disk
DELETE #?.O	Will delete all files ending in .O (object files)
DELETE STUFF/#?	Will delete everything in the STUFF directory but not the directory itself
COPY DF1:#?.C DF0:	Will copy all files ending in .C from DF1 to DF0
COPY V#?.ASM RAM:	Will copy V0.ASM, V1.ASM, VX.ASM, V_.ASM etc. to RAM

Note that use of wildcards is not a feature of the operating system, but only something recognised by the four programs Delete, List, Search and Copy, and that the List program requires you to specify the P or PAT option.

Hopefully this is enough to clarify how wildcards can be used to your advantage from the CLI.

ADDING A DISK DRIVE.

This article is aimed at the "Do It Yourselfer's", who would like to save a bit of money, or add an 80 track DS DD 5.25" disk drive to their Amiga.

How difficult.

We have all heard stories about adding 5.25" drives, or for that matter any drive to the Amiga. In the words of many a politician, "Let me tell you this...." it isn't hard. The only difference between a normal disk drive interface and the Amigas' is that the Amiga expects a "latching" drive motor, (I'll explain this later), it likes to be told through hardware that you've changed the disk, and it uses auto-configuration to find the drive. These tasks are relativly easy to achieve. There is only one slightly difficult thing about adding a drive, finding DB23s (commonly refered to as @#$%!^%$*& connectors). You can't get them so don't try. All that you have to do is modify a DB25 by pulling out two pins and bending the end of the case.

All the signals on the Amigas' DB23 correspond to normal 34way connector signals except the "Diskchange" signal which is unique to the Amiga. The corresponding signal lines and connections are given in table 1. Note that the "Motor on"and "Ready" signals needs modification.

So what's a latching motor.

Basically the Amiga dosen't want to look after holding on seperate motor lines. With most micros, when one disk motor goes on the motors of any other drive in the system go on as well. This is in efficient and wastes the drive motors' life, which is specified as being around half an hour continuous. To get around the problem of having only one "Motor on" line and not turnning on all drives at once, the Amiga asks the drive to turn on its' motor till further notice, ie latch it. To do this it requires that the "Motor on" signal for each drive latch when the drive is selected. IE. When the drive is selected it takes note, (latches/holds/remembers) what the "Motor on" signal was, on or off. To make everyone confused the disk drive in the Amiga doesn't do this. It has a little circuit to do.

Since most disk drives that are available don't latch the motor signal a simple circuit is required for each drive. Figure 1 shows the latch circuit I use. This circuit is capable of driving the "Motor on" signal of one drive, and should be built into the box used for your disk drive.

Diskchange.

Diskchange is a signal that is supplied by the drive to the Amiga that says "hey guys, I don't have any thing in me!!". Rather like baby Magpies squarking, (or my Autobank). This signal is only supplied by the drive when it is selected. When the drives aren't in use the Amiga regularly polls the drives to check their status.

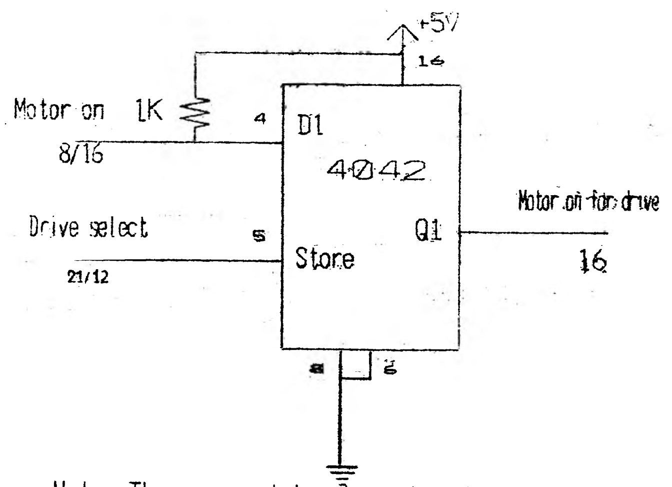
Workbench 1.2 allows you to do this from the CLI using the command DiskChange. The simplest way to get this signal automatically is to incorporate a microswitch in the drive (For 5.25") to sense when it has a disk. The switch should be normally closed, ie when the disk is out of the drive it should connect ground to the disk change line. This will force the disk change signal on. The only drawback with this method is that you should have a disk in every other drive to avoid confusion. If you don't the Amiga will not "load" any drive when it gets a new disk. Note I would not recommend this method, however it has been tried with no ill effects.

To get around this problem you will need a circuit like the one in figure 2. I haven't gotten around to trying this circuit yet, I'll keep you informed about that one. The idea with this circuit is that the diskchange signal is held in a flip-flop and only tells the computer it doesn't have a disk when the drive is selected. It should work, and it can't harm your Amiga. (Note: the hardware manual appendix has a

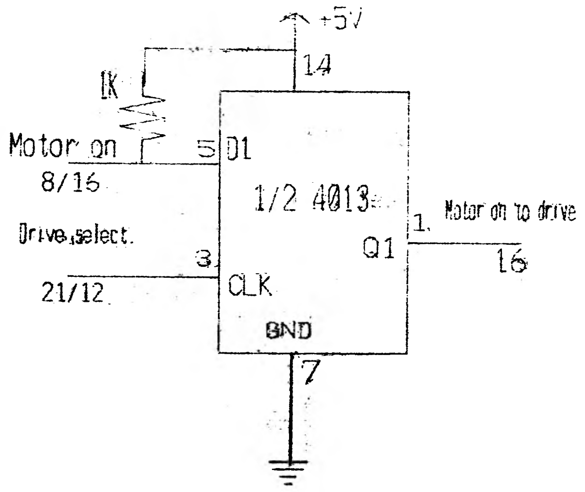
Table 1. Amiga Disk Drive Connector Signals

Amiga	34Way	Description.
1	34	Ready: Indicates drive ready with disk. Also used for drive identification.
2	30	Read data: Data input to the Amiga.
3	any odd.	GND: Ground return
4	any odd.	GND: Ground return
5	any odd.	GND: Ground return
6	any odd.	GND: Ground return
7	any odd.	GND: Ground return
8	16*	Motor on data, clocked into drives motor-on flip flop by the active transision of SELxB. ie. This signal should latch when the drive is selected. (Open collector output from Amiga (OC))
9	14	Drive select 2: Drive select for the second external drive, ie df2: (OC)
10	-	DRESB: Amiga system reset you could reset motor latch and set write protect, not necessary. (OC)
11	(2)	Disk change: you may return this signal on line 2 of the 34way cable as this is usually spare.
12	-	+5V ************************* Avoid using!!!
13	32	Side select: This chooses which side of the disk to read or write from/to.
14	28	Write protect: set by selected WP'ed disk in drive
15	26	Track Zero: asserted when head on track 00.
16	24	Write gate: Write enable to drive (OC)
17	22	Write data: Data from Amiga to drive. (OC)
18	20	Step heads: Selected drive steps its' heads one cylinder in direction of Direction select signal. (OC)
19	18	Direction select: determines above. (OC)
20	6	Drive select 3: df3 select, ie 3rd external drive (OC)
21	12	Drive select 1: df1 select, ie 1st external drive (OC)
22	8	Index: Index pulse for the selected drive.
23	-	+12V ************************* Avoid using.

OC - Open collector output from Amiga.
Although the power rails MAY power one external drive, I'd advise against using it on a 5.25" drive.



Note: This circuit has been tried and tested.



Note: This circuit has not been tested, but should work. It is easier to obtain 4013s. and you can use the other 1/2 for the Disk change circuit.

Figure 1. Motor Latch for Amiga Drives

R. McNeil 11/12/86

beautiful display of contradiction on how disk change works). The drive tells the Amiga that it hasn't been selected since a disk has been put in by not changing the signal till a "Drive select" and "Step" signal have been given. (This is the ticking sound you hear when no disk is present in the drive.)

What drive to use.

Any 5.25", 1MB double sided drive should work with the Amiga. We have tried the Mitsubishi M4852/M4853 1MB half height drive supplied by Rod Irving Electronics and the TEAC drive supplied by Steves. We tried a Mitsubishi 3.5" MF353A-12U drive, but found that its' direction select signal was far to slow for the Amiga. as an aside, we think these drives have been dumped on the Aussy market as they are cheeper than Rods 5.25" drives and have rather slow access times. (Probably good for IBMs.)

Boot up.

On boot up the Amiga goes and checks what drives are hanging on the system. To do this it selects the drive without turnning on the motor. It then expects the drive to reply with its name (eg 11111111 for a 3.5" and 55555555 for a 5.25" drive) through the ready line. Rather like a roll call, if a drive doesn't reply (ie replies with 00000000) it isn't there. A normal drive doesn't have this mode of identification, so we have to do this too. Workbench 1.2 will let you mount another drive from the CLI by typing the command MOUNT followed by the drive (ie df1,df2,df3).

To make boot up automatic you can try two methods both make the Amiga think the drive is a 3.5". With method one you turn on the drives motor with a switch as in figure 3 before booting the workbench. This will make the ready line assert when the drive is selected, which the computer will read as 11111111. The second method is to use the circuit of figure 4 on the ready line between the drive and the Amiga. This senses the motor being off when the drive is selected and then asserts the ready signal with the same result. (Note I haven't tried this (2nd method) either).

Other things.

The only other thing you will have to think about is a Power supply for your drive. You will have to supply both +5 and +12 volts to you drive. The supply must be able to deliver 1 amp into the lot for one drive.

Construction considerations.

The first thing you have to work out is how many drives you'd like and how you're going to box them. Some questions you might like to ask are whether the power supply will fit in the same box as the drive? Am I going to run off the back of an Amiga drive or out of the computer? What drive am I going to use? (Can I afford it?)

If you are going to plug your drive into the back of an existing Amiga external drive you will have to set the drive select on your drive to drive 1. This is because the Amiga drive swaps the drive select lines so that you don't have to change the internal drive select jumpers on the drive. Ie. they don't want you to open their drives.

If you are mixing drives you'll have to work that out what selects go where. The best idea is to put all your Amiga drives first, and then add your own drives starting with the drives jumper on df1 then the next on df2 etc.

Putting it together.

The best way to construct your drive is to mount the drive and circuitry in one box and the power supply in another. The box should have a 34way connector on the back of it. This connector should then go to a board with any of the electronics you want to use on it, and then this board should have a 34way ribbon cable with connector that goes to the drive. This way all the external circuitry is separate to the drive itself. Figure 6 shows a complete circuit connection between the Amiga and your drive. (didn't I say it was easy!?).

I can't really give you any more detail than this as it depends on your own situation, however I can leave you with some warnings and notes.

Do NOT physically disconnect and drive from your Amiga while it is on. Both the Amiga and the drive power should be off or you risk blowing line drivers. It happens (I've had one drives' line driver do this).

The Amiga is well protected from external devices but please be careful when connecting power supply lines. NOTE: The Author will not be held responsible for any damage caused by use of any circuits or information in this article. Oh yeah, GOOD LUCK.

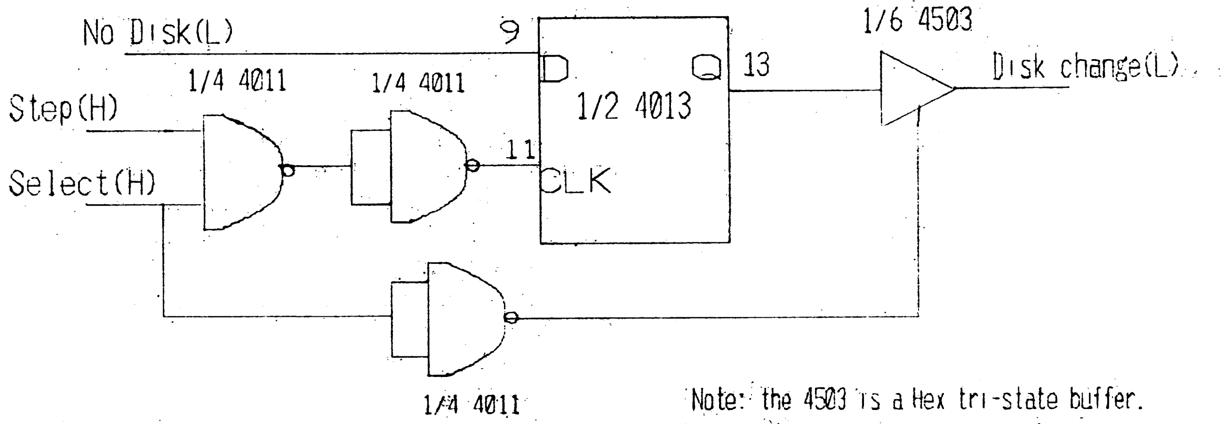
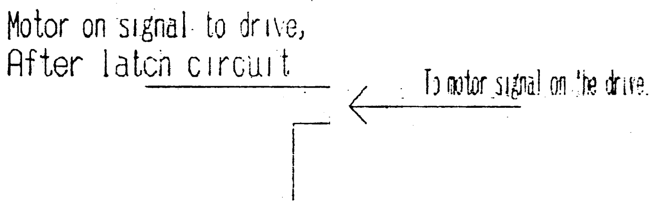


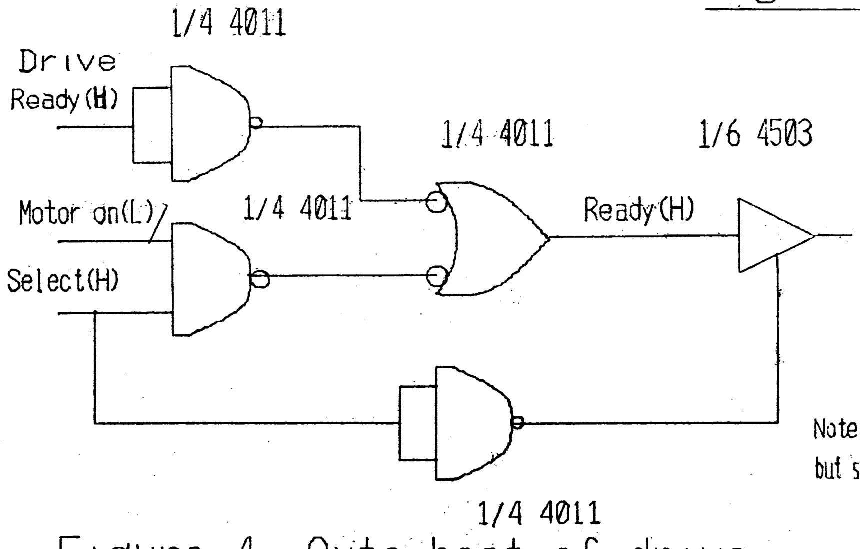
Figure 2. Disk change circuit.

Note: the 4503 is a Hex tri-state buffer. This circuit has not been tried.



Note: This has been tried and works well

Figure3 Boot switch.



Note: This circuit has not been tried. but should help in understanding the problem.

Figure 4. Auto boot of drive.

Spring Meditations

Some Thoughts on a Newly Constructed 5.25" Drive

There is no doubt that using an Amiga with only one drive is somewhat inefficient (I am being kind here), and after the first series of trashed disks resulting from frantic swaps to achieve the most basic of functions, one's thoughts turn in a spring-like euphoria to an external drive. However, the price of an Amiga external drive kept me in a state of frustration and high blood pressure for a further three months, until a week ago when I could endure it no more, and the most cost-efficient way was to build my own from a bare 5.25" drive. The project has consumed about two days and a couple of evenings (which also included rebuilding the power supply when the 1 amp voltage regulator proved to be too small for my particular drive), but now I am the proud father of a chirpy TEAC external drive and all those file-shuffling jobs are now performed with consumate ease. However, the home-brewed drive may not be for everyone, for there are a couple of idiosycrasies with the larger drives which some may find annoying and therefore, perhaps, unacceptable. But if these 'quirks' don't worry you, then go right ahead and save yourself $250 or so.

The first idiosyncracy occurs at boot-up time. With Workbench 1.1 the drive power supply must be turned on, a disk inserted, and the extra motor-switch mounted on the front of the drive (or wherever you've put it) must be turned on BEFORE Kickstart is loaded and remain in this way until Workbench has finished loading, for only then will the initialisation procedure actually mount DF1. Once you are at the CLI or Workbench screen, the motor switch can be turned off and you don't have to worry about it again until you re-boot. (It is important that you don't forget to turn the motor-switch off, because otherwise the disk drive motor is running continuously - a state which it was not designed for.) Workbench 1.2 only requires the motor-switch to be on for the Workbench disk, and even if you have not mounted DF1 with this procedure, a new "MOUNT" command in 1.2, used from the CLI, should enable you to get Workbench to recognise the existence of the drive. If you do not wish to use DF1 then simply leave it turned off BUT you must leave a disk in the drive, and this leads to the second idiosyncracy.

To enable the system to know when a disk has been changed, most 5.25" drives require the addition of a micro-switch in the disk-path to tell the system when a disk is removed and another inserted. This results in the system ignoring disk changes in the internal drive if there is no disk in the external drive - and it makes no difference if DF1 is powered up or not, nor whether DF1 is mounted or not. There MUST be a disk in DF1 at all times. An option around this problem is to not include a disk-path micro-switch on the drive and then to use the new 'DISKCHANGE' command in 1.2 whenever a new disk is to be inserted into DF1. However, this would necessitate constant access to the CLI to execute the command - a restriction which I personally found unacceptable.

At this point you may be wondering if the 5.25" drives are worth the trouble, but let me assure you, they are. The two problems described above are very easily endured when compared to the frustrations of one-drive operations. In fact there are only two things to remember:

(1) turn ON the the motor-switch while booting

(2) When the system has booted turn the motor-on switch back to its normal position

Happy disk shuffling... Keith Elwell-Gavins

Keyboard Hints -- Craig Fisher

The following is a collection of hints and tips regarding use of the keyboard. In the following, <LEFT-A> means "hold down the left 'Amiga Key'", just like the Shift key with punctuation.

Special Keyboard Codes:

<LEFT-A><N> will bring your WORKBENCH screen to the front; and
<LEFT-A><M> will push your WORKBENCH screen to the back.

This is especially useful when you have a screen with no depth gadgets. Note that this operates only on screens, not windows.

Pressing <LEFT-ALT><LEFT-A> is the same as pressing the left mouse button.
Pressing <RIGHT-ALT><RIGHT-A> is the same as pressing the right mouse button.
Pressing either AMIGA key and holding down an arrow key will move your mouse pointer as if you were moving the mouse.

The above are all standard key combinations recognised by Intuition (see page 208 in the Intuition Reference Manual [Addison-Wesley]), this means that they won't work if Intuition is not active (e.g. with self-booting games).

Reading the Keyboard from AmigaBasic:

The following is a list of codes returned by special keys on the Amiga keyboard when using the INKEY$ function of AmigaBasic. All other keys return standard ASCII codes. (An ASCII chart is given in the AmigaBasic Manual)

KEY	CODE	KEY	CODE
<UP-ARROW>	= 28	<F1>	= 129
<DOWN-ARROW>	= 29	<F2>	= 130
<RIGHT-ARROW>	= 30	:	:
<LEFT-ARROW>	= 31	:	:
<HELP>	=139	<F10>	= 138

Pressing <CONTROL> and an alphabetic key (A-Z) will return a character from 1 to 26. e.g. Pressing <CONTROL><H> returns the value 8 which is the ASCII code for backspace and is the same as pressing the <BACKSPACE> key.

PHOTOGRAPHING THE AMIGA DISPLAY

Scott Moorhen

The ability to photograph the Amiga display can be quite a useful asset, not only for recreation but also for academic or professional work. It allows you to convey all the details, most importantly the colours used, to maximise the effectiveness of your diagram or picture.

Those of you with full colour printers need read no further but for the rest of you I will now give you a guide for successful Amiga photography, based on the experience I have gained through experimentation on a Saturday afternoon.

The equipment required is basically a 35mm SLR camera and a stable platform. An SLR camera is required to give a 'through the lens' view which will greatly simplify the process of getting a good, sharp picture. I have not tried to use any other type of camera but I would not imagine that anything simpler than an SLR camera (ie an instamatic or something similar), would make life very easy at all.

A platform, preferably a tripod or similar, is needed because, as I will explain, the exposure time is one second and any movement will destroy your brilliant focussing efforts.

As a subject, I selected various Dpaint pictures since these are well known amongst users and hence provide a good reference.

The monitor itself acts as the light source in this procedure and hence room lighting should be subdued. If there is a window behind your monitor the curtains should be closed or the light blocked off in some way. Basically you should have no glare on the monitor from any external light source.

The monitor settings for my experiments were centred ie unbiassed and if you are going to use my suggested aperture settings then you should also centre the settings on your monitor.

The camera should be placed in such a way as to give a good full picture, ie the border of the picture tube should be as large as possible in the viewfinder. With the 50mm lens I was using, the distance turned out to be about 1m.

The most crucial part of this procedure is finding the appropriate aperture settings for the picture. This was a process of trial and error but here is a table of results from my experience.

Picture	Setting
Venus	f8 - perfect f4 - a touch too bright
Title (paintcan)	f11 - a bit dark f8 - a bit bright
King Tut	f5.6 - perfect f8 - good
Graph	f11 - good f8 - good f5.6- colours washed out

All of these shots were taken with an exposure time of 1 second. I do not have an infinite film budget so extensive variation of these settings did not take place.

The pictures give a reasonable cross section of light intensisties. Your pictures can be compared with these as a rough gauge to intensities and hence put you in the ballpark for the correct aperture setting.

As a final point I would like to emphasise that ultra high quality equipment is not essential, for instance I was using 'replacement' 100ASA 35mm colour print film with the processing being performed by a local processing house. The results I obtained were very pleasing and I would recommend that curious people should try to get good results for themselves, without hesitation.

I would be interesed in getting any further information on other peoples experiences as I have a strong interest in this area.

Food for thought - I have photographed static displays, I wonder what effects animated displays could give ? Hmmm...

Review of FileZap 1.0 -- Craig Fisher

This is a brief review of the Public Domain program 'FileZap 1.0'. This program is one of the many Amiga public domain programs being distributed on what are known as the 'Fish Disks'. The Fish disks are a series of disks (now up to 35) of programs compiled by an American named Fred Fish (yes, that is his real name). FileZap can be found on FishDisk 14. The Fish Disks can be copied from Steve's, the Society or you can download some of the programs that have been placed onto the bulletin board (see the front page for BBS details). Hopefully we will have a series of reviews of public domain software in BECAUSE to let you know what is available, what it does and just get you interested. The public domain software available for the Amiga is a great resource which every Amiga owner should be using to their advantage. Most of the programs come with source code so you can use them as examples for your own programming or modify them to suit your needs. Most of the programs, including FileZap, are written in C with a few written in assembler and Modula-2.

Well, onto the review... FileZap is one of the more useful programs I have come across amongst the public domain software. It is a utility to allow you to modify any file. Two compiled versions are supplied - one compiled by the Lattice C compiler and the other by the Manx C compiler. The Manx compiled version is smaller but doesn't seem to like filenames with spaces in them. FileZap is invoked by the command "filezap filename". Files are displayed 256 bytes (one record) at a time, in both hexadecimal and ASCII form at once (similar to TYPEing a file with the H option from CLI).

You can move through the file by jumping forwards or backwards (F or B) by one record, by jumping to the first or last record (S or E) in the file or by jumping to a specific record within the file. The arrow keys are used to move your cursor up, down, left and right within an individual record. Any byte can be changed to a new value entered in either ASCII where you just type the appropriate character or in HEX where you enter two hex digits. After entering any changes you must press U to update the record in the file. You can then press CTRL-C to escape from the program with your file modified.

FileZap is not at all fancy or sophisticated but does do what it was designed for. If you are writing programs which create or use files you may find it handy to set up or examine files used by your program (I have used it many times for this purpose). I also found it handy to be able to modify the 'system-setup' file to select the MIDI baud rate when V1.1 of preferences wouldn't allow it to be selected. It can also be used as an alternative to the list command to examine files - especially handy if you want to jump about or look at the last record of a very large file.

Verdict: A very useful utility to add to your library.

Performance:	*****
Documentation:	***
Ease of use:	****
Value for money:	***** (free)
Reality Coeff.:	0.99
Correctness: (lack of bugs)	****

Review of PopCLI -- Craig Fisher

PopCli is another of the many Amiga public domain programs on the 'Fish Disks' (see FileZap review). This is a neat little utility consuming only a few K of memory which serves two purposes. The first is the ability to "Pop" up a new CLI whenever you like - from any program that doesn't completely take over the machine - by pressing <LEFT-AMIGA><ESCAPE>. You can pop up as many CLI's as you like and just get rid of them using the EndCLI command. Instead of bringing up a NewCLI you can make PopCli start up some other command such as Dir whenever it is invoked. Normally you start PopCli by typing 'run popcli', which will just use the default parameters - i.e. start a new CLI when invoked. Using 'run popcli list' would make PopCli execute the List command whenever you invoked it with <LEFT-AMIGA><ESCAPE>.

The second purpose of PopCli is to increase the life of your monitor. It does this by switching off the monitor after a certain period of inactivity - no input from user. You can set the amount of inactive time you want to pass before PopCli will cut in and black out the display as a command line parameter when it is started up. Here is an example:

```
run >nil: PopCLI >nil: 300 NewCLI "con:0/10/640/120/PopCLI Window"
```

This will close down the screen after five minutes (300 seconds), and when Left-A/Esc is pressed, will load a new CLI in a window in the top half of the screen.

Blacking out the display is supposedly better than switching your monitor on and off which causes wear on the power supply. Pressing any key on the keyboard or just moving the mouse will instantly restore your display to what it was before it switched off.

Now if you find yourself stuck in the middle of some program which you don't want to exit from but want to execute some CLI command you can just invoke PopCli to give you a new CLI. Every time you don't press a key on the keyboard or move the mouse for a given amount of time your monitor will switch off and its life expectancy will be slightly increased.

Verdict: Yet another very useful utility to keep in your C: directory.

Performance:	*****
Documentation:	*****
Ease of use:	*****
Value for money:	*****
Reality Coeff.:	0.99
Correctness: (lack of bugs)	*****

Guru Meditation

I decided that a question and answer section in the newsletter would be a good way of letting people air their problems and share answers to the ones that we solve, hopefully all.

I will try to collect a few questions for each issue either from meetings, the BBS or my own queries. If possible I will print answers along with the questions, otherwise answers will (hopefully) appear in the following issue, when people will have had a chance to read the newsletter and solve them.

We'll start the ball rolling (bouncing?) this month with a couple of questions of our own:

Topic: Intuitions Proportional Gadgets
Question: When creating a proportional gadget under Intuition you must supply an uninitialised image structure for the gadget rendering and Intuition fills it in for you. Is there any way of telling it what colours you want your proportional gadget to be rendered in?

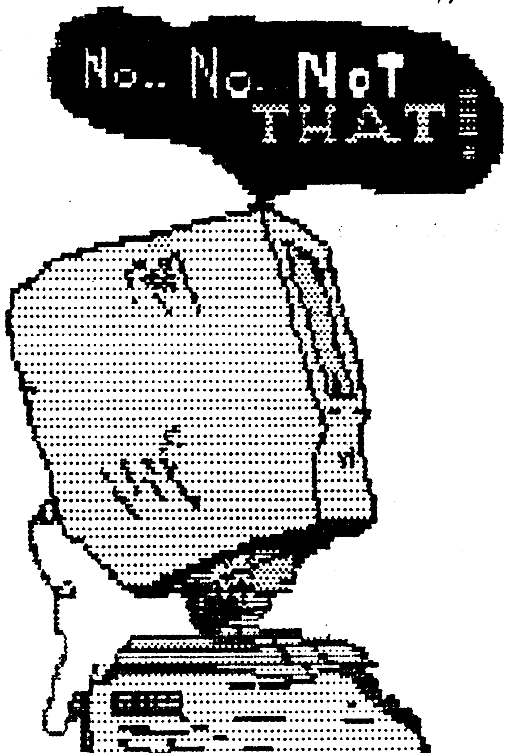
Topic: Keycodes
Question: How does one convert a raw keycode to an ASCII value? (Without using a look-up table!) e.g. using a keymap.

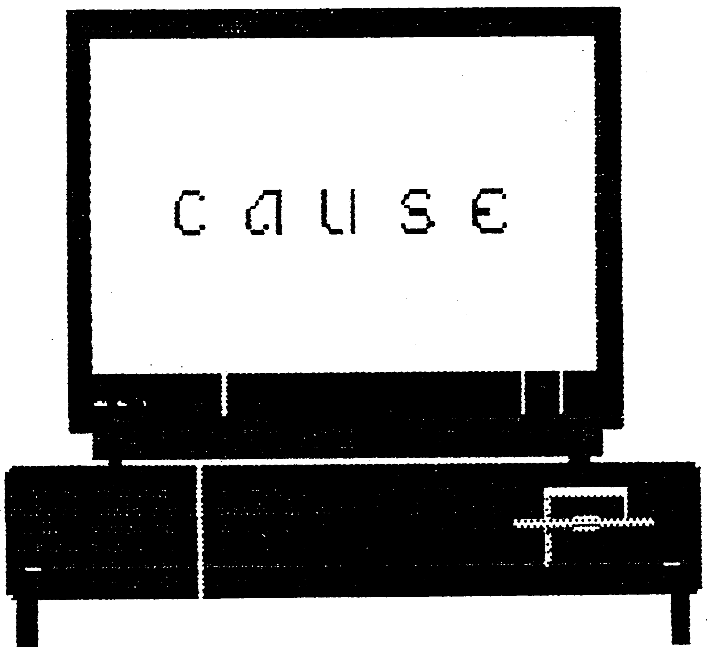
Topic: Hardware
Question: When is a hard disk going to be available?

Answer: Real Soon Now; The Melbourne Amiga User's Group has designed a very inexpensive, extremely simple Hard Disk interface. Reality Coefficient of 0.49 -- I will be following this up PERSONALLY -- Wayne.

Topic: Sound
Question: Does anyone know the format of the sampled sound files for MusiCraft SoundScape, and WorkBenchDemo?

Topic: Disk Errors
Question: Has anyone found a reliable way to un-corrupt a disk? Current methods known include the 1.2 DiskDoctor (which CAN be very violent to the disk), DiskSalv, and last (but not least) Format...





The Philosophy Page

This issue : FRIEDRICH NIETZSCHE (1844-1900)

1. Even the bravest of us rarely has the courage for what he really *knows*.

2. He who does not know how to put his will into things at least puts a *meaning* into them: that is, he believes there is a will in them already (principle of 'belief').

3. One seldom commits only one rash act. In the first rash act one always does too much. For just that reason one usually commits a second - and then one does too little ...

4. When it is trodden on a worm will curl up. That is prudent. It thereby reduces the chance of being trodden on again. In the language of morals: *humility.* -

5. You run on *ahead*? - Do you do so as a herdsman? or as an exception? A third possibility would be as a deserter... *First* question of conscience.

6. Are you genuine? or only an actor? A representative? or that itself which is represented? - Finally you are no more than an imitation of an actor....*Second* question of conscience.

7. Are you one who looks on? or who sets to work? - or who looks away, turns aside....*Third* question of conscience.

8. Do you want to accompany? or go on ahead? or go off alone?... One must know *what* one wants and *that* one wants .
   - *Fourth* question of conscience.

9. Formula for my happiness: a Yes, a No, a straight line, a *goal...*

Donated by: *Twilight of the Idols or How to Philosophize with a Hammer* , Penguin Books Ltd., 1979 Ed.

BECAUSE

36 Ambalindum St.
HAWKER A.C.T. 2614
Ph. (062) 54 6033

Bulletin of the
Canberra Amiga
Users Society